

**Examenul de bacalaureat național 2020**  
**Proba E. d)**  
**Informatică**  
**Limbajul C/C++**

Testul 13

*Filieră teoretică, profil real, specializare științe ale naturii*

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (bold), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.

**SUBIECTUL I** **(20 de puncte)**

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

- Indicați o expresie C/C++ care are valoarea 1 dacă și numai dacă numărul natural memorat în variabila întregă  $n$  este divizibil cu 4 și cu 5.
  - $!(n\%4!=0 \ || \ n\%5!=0)$
  - $!(n/4==1 \ || \ n/5!=0)$
  - $n\%4==0 \ \&\& \ !(n\%5==0)$
  - $n/4==0 \ \&\& \ !(n/5==0)$
- Pentru a verifica dacă într-un tablou unidimensional există elementul cu valoarea  $x=4$ , se aplică metoda căutării binare, iar succesiunea de elemente a căror valoare se compară cu  $x$  pe parcursul aplicării metodei este 14, 8, 4. Elementele tabloului pot fi:
  - (48, 14, 9, 8, 7, 4, 2)
  - (14, 14, 8, 8, 4, 4)
  - (14, 8, 4, 3, 2, 0)
  - (4, 8, 9, 14, 16, 24, 48)
- Tablourile unidimensionale  $A$  și  $B$  au elementele:  $A=(50, 36, 27, 20, 2)$ , iar  $B=(63, 45, 8, 5, 3)$ . În urma interclasării lor în ordine crescătoare se obține tabloul cu elementele:
  - (2, 3, 5, 8, 20, 27, 36, 45, 50, 63)
  - (2, 5, 8, 27, 50)
  - (50, 36, 27, 20, 2, 63, 45, 8, 5, 3)
  - (50, 63, 36, 45, 8, 27, 5, 20, 2, 3)
- O expresie C/C++ care are valoarea 1.0 este:
  - `ceil(-0.2080)`
  - `ceil(0.8020)`
  - `floor(-0.2080)`
  - `floor(0.8020)`
- În secvența de instrucțiuni alăturată, toate variabilele sunt de tip întreg. O expresie care poate înlocui punctele de suspensie astfel încât, în urma executării secvenței, variabila  $x$  să aibă o valoare egală cu cel mai mare divizor comun al numerelor 2020 și 2200 este:

	<code>x=2020; y=2200;</code> <code>while (y!=0)</code> <code>{ z=...; x=y; y=z; }</code>
<ol style="list-style-type: none"><li><math>x\%2</math></li><li><math>x/2</math></li><li><math>x\%y</math></li><li><math>x/y</math></li></ol>	

**SUBIECTUL al II-lea** **(40 de puncte)**

- Algoritmul alăturat este reprezentat în pseudocod.**  
S-a notat cu  $a\%b$  restul împărțirii numărului natural  $a$  la numărul natural nenul  $b$  și cu  $[c]$  partea întreagă a numărului real  $c$ .
  - Scrieți valoarea care se afișează în urma executării algoritmului dacă se citesc, în această ordine, numerele 12345, 25, 2070, 49, 270135, 21. **(6p.)**
  - Dacă pentru  $n$  se citește numărul 100, scrieți un set de date din intervalul  $[10^3, 10^4)$  care pot fi citite în continuare, astfel încât, în urma executării algoritmului, să se afișeze 10. **(6p.)**
  - Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
  - Scrieți în pseudocod un algoritm, echivalent cu cel dat, care să conțină o singură instrucțiune repetitivă. **(6p.)**

	<code>citește n (număr natural)</code> <code>p←1; m←0; k←0</code> <code>┌cât timp n≠0 execută</code> <code>│ citește x (număr natural)</code> <code>│ i←k</code> <code>└cât timp i≠0 execută</code> <code>│ x←[x/10]; i←i-1</code> <code>└─</code> <code>┌dacă x=0 atunci c←n%10</code> <code>│ altfel c←x%10</code> <code>└─</code> <code>m←c*p+m; n←[n/10]</code> <code>p←p*10; k←k+1</code> <code>└─</code> <code>scrie m</code>
--	---

2. Variabila întreagă **raza** memorează raza unui cerc, iar variabilele reale **centruX** și **centruY** memorează coordonatele reale (abscisa și ordonata), în planul **xOy**, ale centrului unui cerc. Declarați corespunzător variabilele **centruX** și **centruY** și scrieți o secvență de instrucțiuni în urma executării căreia să se afișeze pe ecran mesajul **Unitar** dacă cercul precizat are centrul în originea sistemului de coordonate și raza egală cu 1, sau mesajul **Altceva** în caz contrar. (6p.)
3. Variabila **i** este de tip întreg, iar celelalte variabile sunt de tip **char**. Scrieți ce se afișează în urma executării secvenței alăturate. (6p.)
- ```
c1='s';c2='o';c3='u';  
cout<<c2<<c3<<endl; | printf("%c%c\n",c2,c3);  
for(i=0;i<4;i++)  
{ c=c1+(i-1)*(1-i%2)+3*(2*i/3-1)*(i%2);  
  if(i==0) c1=c;  
  cout<<c; | printf("%c",c);  
}
```

**SUBIECTUL al III-lea**

**(30 de puncte)**

1. Se citește un număr natural **n** ( $n \geq 2$ ) și se cere să se scrie, în această ordine, numerele **d** și **p**, reprezentând divizorul prim, **d**, care apare la cea mai mică putere, **p**, în descompunerea în factori primi a lui **n**; dacă există mai mulți astfel de divizori se afișează cel mai mic dintre ei. Scrieți, în pseudocod, algoritmul de rezolvare a problemei enunțate. **Exemplu:** dacă  $n=10780$ , atunci se scriu numerele 5 1 ( $10780=2^2 \cdot 5 \cdot 7^2 \cdot 11$ ). (10p.)
2. Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul  $[2, 20]$ , **n** și **k**, și construiește în memorie un tablou unidimensional cu  $n \cdot k$  elemente astfel: tabloul memorează un șir crescător de termeni cu proprietatea că primul termen este **n**, fiecare valoare apare în șir de exact **k** ori și oricare doi termeni alăturați au valori egale sau consecutive. Programul afișează pe ecran tabloul construit, cu valorile separate prin câte un spațiu. **Exemplu:** dacă  $n=3$  și  $k=4$ , se obține tabloul alăturat. (10p.)
- ```
3 3 3 3 4 4 4 4 5 5 5 5 6 6 6 6
```
3. Șirul de mai jos este definit astfel:  $f_1=1$ ,  $f_n=2 \cdot f_{n-1}$  (unde **n** este un număr natural  $n \geq 2$ ).  
1, 2, 4, 8, 16, 32, ...  
Se citește de la tastatură un număr natural **x** ( $x \leq 10^9$ ), valoare a unui termen al șirului dat, și se cere să se scrie în fișierul text **bac.txt**, în ordine descrescătoare, separați prin câte un spațiu, toți termenii șirului care sunt mai mici sau egali cu **x**. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare. **Exemplu:** dacă se citește numărul 16  
fișierul **bac.txt** conține numerele 16 8 4 2 1  
a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)  
b. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)

Examenul de bacalaureat național 2020  
Proba E. d)  
Informatică

BAREM DE EVALUARE ȘI DE NOTARE  
(comun pentru limbajele C/C++ și Pascal)

Testul 13

Filieră teoretică, profil real, specializare științe ale naturii

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.

SUBIECTUL I

(20 de puncte)

1a 2d 3a 4b 5c	5x4p.
----------------	-------

SUBIECTUL al II - lea

(40 de puncte)

1.	a) Răspuns corect: 10375	6p.	
	b) Pentru răspuns corect	6p.	Se acordă câte 2p. pentru fiecare număr conform cerinței (un set are trei numere din intervalul cerut, a, b și c unde a are cifra unităților 0, b are cifra zecilor 1, iar c are cifra miilor 0).
	c) Pentru program corect -declarare variabile -citire date -afișare date -instrucțiune de decizie -instrucțiuni repetitive (*) -atribuiri -corectitudine globală a programului <sup>1)</sup>	10p. 1p. 1p. 1p. 2p. 3p. 1p. 1p.	(*) Se acordă numai 2p. dacă doar una dintre instrucțiunile repetitive este conform cerinței.
	d) Pentru algoritm pseudocod corect -echivalență a prelucrării realizate, conform cerinței (*) -corectitudine globală a algoritmului <sup>1)</sup>	6p. 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinței, principial corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă corectă de structură repetitivă conform cerinței.
2.	Pentru rezolvare corectă -declarare a variabilelor conform cerinței -afișare a mesajelor (*) -corectitudine globală a secvenței <sup>1)</sup>	6p. 1p. 4p. 1p.	(*) Se acordă câte 2p. pentru fiecare mesaj conform cerinței.
3.	Răspuns corect: ou rosu	6p.	Se acordă câte 1p. pentru fiecare literă afișată conform cerinței.

SUBIECTUL al III - lea

(30 de puncte)

1.	Pentru algoritm corect -citire a datelor -determinare a valorilor cerute (*) -scriere a datelor -scriere principial corectă a structurilor de control, corectitudine globală a algoritmului <sup>1)</sup> (**)	10p. 1p. 6p. 1p. 2p.	(*) Se acordă câte 1p. pentru fiecare aspect al cerinței (identificare a unui divizor, identificare a unui divizor prim, divizori suport, putere a unui divizor, tratare caz divizori cu aceeași putere, algoritm principial corect de determinare a unei valori minime). (**) Se va puncta orice formă corectă de structură repetitivă sau decizională.
2.	Pentru program corect -declarare a unei variabile care să memoreze un tablou unidimensional -citire a datelor -construire a tabloului conform cerinței (*) -afișarea a datelor în formatul cerut -declarare a variabilelor simple, corectitudine globală a programului <sup>1)</sup>	10p. 1p. 1p. 6p. 1p. 1p.	(*) Se acordă câte 2p. pentru fiecare aspect specific (primul element, elemente egale, valori suport) conform cerinței.

<b>3.</b>	<b>a) Pentru răspuns corect</b> -coerență a descrierii algoritmului (*) -justificare a elementelor de eficiență	<b>2p.</b> 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare.
	<b>b) Pentru program corect</b> -operații cu fișiere: declarare, pregătire în vederea scrierii, scriere în fișier -determinare a valorilor cerute (*),(**) -utilizare a unui algoritm eficient (***) -declarare a variabilelor, citire a datelor, corectitudine globală a programului <sup>1)</sup>	<b>8p.</b> 1p. 5p. 1p. 1p.	(***) Se acordă punctajul numai pentru un algoritm liniar care utilizează eficient memoria. O soluție posibilă generează termenii șirului deducând forma generală a termenului curent și actualizând, la fiecare pas, valoarea acestuia: dacă crt, x sunt doi termeni aflați pe poziții consecutive în șir, în această ordine, atunci $crt=x/2$ ; valorile variabilelor crt și x se actualizează la fiecare pas.

<sup>1)</sup> Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.